IN THE CLAIMS:

Please amend the claims as follows:

1. (Currently Amended) A method for buffering data produced by ~~in~~ a computer graphics pipeline, comprising:

~~(a)~~ producing graphics floating point data in a graphics pipeline;

~~(b)~~ operating on the graphics floating point data in the graphics pipeline; and

~~(c)~~ storing the graphics floating point data to a buffer ~~in the graphics pipeline~~;

~~(d)~~ wherein the graphics floating point data includes fragment data received from a rasterizer that is read and stored in an unclamped format dictated by a graphics application program interface for increasing a parameter selected from the group consisting of a precision and a range of the graphics floating point data.

Sub B1

2.-3. (Cancelled)

a1 cont

4. (Currently Amended) The method as recited in claim 1~~2~~, wherein the fragment data includes color data.

5. (Currently Amended) The method as recited in claim 1~~2~~, wherein the fragment data includes depth data.

6. (Original) The method as recited in claim 1, wherein the graphics floating point data is only constrained by an underlying data type.

7. (Original) The method as recited in claim 1, wherein the buffer serves as a texture map.

8. (Currently Amended) A computer program product for buffering data produced by ~~in~~ a computer graphics pipeline, comprising:

(a) computer code for producing graphics floating point data in a graphics pipeline;

(b) computer code for operating on the graphics floating point data in the graphics pipeline; and

(c) computer code for storing the graphics floating point data to a buffer ~~in the graphics pipeline~~;

(d) wherein the graphics floating point data includes fragment data received from a rasterizer that is read and stored in an unclamped format dictated by a graphics application program interface for increasing a parameter selected from the group consisting of a precision and a range of the graphics floating point data.

Sub B1

a1 cont

9.-10. (Canceled)

11. (Currently Amended) The computer program product as recited in claim 8~~9~~, wherein the fragment data includes color data.

12. (Currently Amended) The computer program product as recited in claim 8~~9~~, wherein the fragment data includes depth data.

13. (Original) The computer program product as recited in claim 8, wherein the graphics floating point data is only constrained by an underlying data type.

14. (Original) The computer program product as recited in claim 8, wherein the buffer serves as a texture map.

15. (Currently Amended) A system for buffering data produced by ~~in~~ a computer graphics pipeline, comprising:

(a) logic for producing graphics floating point data in a graphics pipeline;

(b) logic for operating on the graphics floating point data in the graphics pipeline; and

(c) logic for storing the graphics floating point data to a buffer ~~in the graphics pipeline~~;

(d) wherein the graphics floating point data includes fragment data received from a rasterizer that is read and stored in an unclamped format dictated by a graphics application program interface for increasing a parameter selected from the group consisting of a precision and a range of the graphics floating point data.

Sub B1

16. (Currently Amended) A buffering apparatus ~~in a computer graphics pipeline~~, comprising:

(a) a buffer capable of storing graphics floating point data produced by ~~in~~ a graphics pipeline;

(b) wherein the graphics floating point data includes fragment data received from a rasterizer that is stored in an unclamped format dictated by a graphics application program interface for increasing a parameter selected from the group consisting of a precision and a range of the graphics floating point data.

a' cont

17. (Currently Amended) A system for buffering data produced by ~~in~~ a computer graphics pipeline, comprising:

(a) means for producing graphics floating point data in a graphics pipeline;

(b) means for operating on the graphics floating point data in the graphics pipeline; and

(c) means for storing the graphics floating point data to a buffer ~~in the graphics pipeline~~;

(d) wherein the graphics floating point data includes fragment data received from a rasterizer that is read and stored in an unclamped format dictated by a graphics application program interface for increasing a parameter selected from the group consisting of a precision and a range of the graphics floating point data.

18. (Currently Amended) A method for buffering data produced by ~~in~~ a computer graphics pipeline, comprising:

(a) producing graphics floating point data in a graphics pipeline;

(b) operating on the graphics floating point data in the graphics pipeline; and

(c) storing the graphics floating point data to a buffer ~~in the graphics pipeline~~;

(d) wherein the buffer serves as a texture map.

Sub B1

19. (Currently Amended) A buffering apparatus ~~in a computer graphics pipeline~~, comprising:

(a) a buffer capable of storing graphics floating point data produced by ~~in~~ a graphics pipeline;

(b) wherein the buffer serves as a texture map.

a' cont

20. (Currently Amended) A method for buffering data during multi-pass rendering ~~in a computer graphics pipeline~~, comprising:

(a) operating on graphics floating point data during a rendering pass in a graphics pipeline;

(b) reading the graphics floating point data from a buffer during the rendering pass ~~in the graphics pipeline~~;

(c) storing the graphics floating point data to the buffer during the rendering pass ~~in the graphics pipeline~~; and

(d) repeating (a) – (c) during additional rendering passes utilizing results of a previous rendering pass.

21. (Original)The method as recited in claim 20, wherein the operating includes deferred shading.

22. (Currently Amended) A method for buffering data produced by ~~in~~ a computer graphics pipeline, comprising:

~~(a)~~ producing graphics floating point data in a graphics pipeline;

~~(b)~~ packing the graphics floating point data in the graphics pipeline; and

~~(c)~~ storing the graphics floating point data to a buffer ~~in the graphics pipeline~~;

wherein the packing facilitates storage of at least two quantities in a single buffer in a single pass.

Sub B1

a'cont

23. (Currently Amended) A method for buffering data produced by ~~in~~ a computer graphics pipeline, comprising:

~~(a)~~ producing graphics floating point data in a graphics pipeline;

~~(b)~~ unpacking the graphics floating point data in the graphics pipeline; and

~~(c)~~ operating on the unpacked graphics floating point data in the graphics pipeline;

wherein the unpacking facilitates storage of at least two quantities in a single buffer in a single pass.

24. (Currently Amended) A method for buffering data produced by ~~in~~ a computer graphics pipeline, comprising:

~~(a)~~ operating on graphics floating point data in a graphics pipeline;

~~(b)~~ producing the graphics floating point data in the graphics pipeline;

determining whether the graphics pipeline is operating in a programmable mode utilizing a command associated with a graphics application program interface;

if it is determined that the graphics pipeline is not operating in the programmable mode, performing standard graphics application program interface operations on the graphics floating point data; and

if it is determined that the graphics pipeline is operating in the programmable mode:

~~(c)~~ storing the graphics floating point data to a frame buffer ~~in the graphics pipeline;~~, ~~(d)~~

wherein the graphics floating point data includes fragment data received from a rasterizer that is read and stored in an unclamped format dictated by a graphics application program interface extension for increasing a parameter selected from the group consisting of a precision and a range of the graphics floating point data.

Sub B1 concl.

25. (New) The buffering apparatus as recited in claim 19, wherein the buffer serves as the texture map by using previous rendering results via an extension of an application program interface.

26. (New) The method as recited in claim 1, wherein the buffer includes a frame buffer.